

Claims

We claim:

1. A development tool for a network based data processing system, the tool comprising a document type developer for assembling document types from prestored object components and associating with each document type a set of states, each state being definitive of at least one document property, said document types being usable to instantiate documents to form a data processing system, said document types being usable to provide said data processing system with state assignment functionality, for assigning to each instantiation successive ones of said set of states and thus applying said respective property to said instantiation, to thereby successively define for said document type instantiation within said data processing system a succession of properties during a life cycle of said document type instantiation.
2. A development tool according to claim 1, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said set of states includes at least one non-persistent state for attributing to document type instantiation stages that are not to be saved within the data processing system.
3. A development tool according to claim 2, wherein said at least one non-persistent state is attributable to a document type instantiation outgoing stage.

4. A development tool according to claim 3, wherein a persistent state is alternately attributable to a document type instantiation outgoing stage that is to be saved within the data processing system.
5. A development tool according to claim 4, said persistent state being effective to cause generation of said outgoing stage to be delayed.
6. A development tool according to claim 5, wherein said delay is set to depend on system load.
7. A development tool according to claim 1, wherein said set of states includes at least one initiating state for assigning to a document type instantiation during creation of said document type instantiation.
8. A development tool according to claim 1, wherein a document type instantiation comprises a document type indicator for indicating a document type of said instantiation.
9. A development tool according to claim 8, operable to set for each data processing system a feature of placing said document type indicator in a document header.

10. A development tool according to claim 1, wherein at least one of said states has a property of check-in check-out control, thereby to restrict editing level access to a single user at any one time.
11. A development tool according to claim 10, wherein said check-in check-out control comprises a time limitation for any given user.
12. A development tool according to claim 1, operable to set for each data processing system a feature of prompting for saving document history each time an instantiation of a document type is assigned another one of said states.
13. A development tool according to claim 1, operable to set for a document type a feature of saving document history each time an instantiation of said document type is assigned another one of said states.
14. A development tool according to claim 1, operable to set for a document type instantiation a feature of saving document history each time said document type instantiation is assigned another one of said states.
15. A development tool according to claim 1, comprising owner allocation functionality for allocating an owner to each document type instantiation.

16. A development tool according to claim 15, said owner allocation functionality being operable to set for each data processing system a feature of placing an indication of said allocated owner in a document header.
17. A development tool according to claim 1, comprising global document identification functionality operable to set for each data processing system a feature of allocating to each document type instantiation a unique global identifier.
18. A development tool according to claim 17, said global document identification feature comprising functionality to place said unique global identifier for each document type instantiation in a document header.
19. A development tool according to claim 1, comprising current state indication functionality operable to set for each data processing system a feature of placing a current one of said set of states of a document type instantiation in a document header.
20. A development tool according to claim 1, comprising state ordering functionality operable to set for at least one document type a defined order of succession amongst said set of states.

21. A development tool according to claim 1, said state assignment functionality being operable to depend upon external input to an instantiation of a document type.
22. A development tool according to claim 1, said state assignment functionality being operable to depend upon a property of a document type instantiation.
23. A development tool according to claim 1, said state assignment functionality being operable to alter a property of a document type instantiation.
24. A development tool according to claim 1, wherein a document security property is definable via said states.
25. A development tool according to claim 24, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said document security property comprises a requirement for a document type instantiation to acquire at least one digital signature when passing through at least one predetermined stage.
26. A development tool according to claim 25, wherein said digital signature is acquirable from a user.

27. A development tool according to claim 25, wherein said digital signature is acquirable from said data processing system.
28. A development tool according to claim 24, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said document security property comprises a requirement for a document type instantiation to have at least one digital signature from a previous stage before being assigned a state attributable to a new stage.
29. A development tool according to claim 28, wherein said digital signature is acquirable from a user.
30. A development tool according to claim 28, wherein said digital signature is acquirable from said data processing system.
31. A development tool according to claim 24, wherein said document type developer is operable to provide functionality for defining said security property at a document type level.
32. A development tool according to claim 15, wherein said properties definable via said states comprise document access permission.

33. A development tool according to claim 15, wherein said owner is exchangeable with a change in state.

34. A development tool according to claim 1, wherein said properties definable via said states comprise document access permission.

35. A development tool according to claim 34, wherein document access permission is definable for a given user.

36. A development tool according to claim 35, wherein said access comprises deleting a document type instantiation.

37. A development tool according to claim 35, wherein said access comprises reading a document type instantiation.

38. A development tool according to claim 35, wherein said access comprises changing a property of a document type instantiation.

39. A development tool according to claim 1, wherein a component protocol is one of a group of protocols comprising: ActiveX, COM, and COM+.

40. A network based data processing system for processing data arranged in documents, wherein said documents are instantiations of document types

obtained from a document type library, said document types having a set of states, each state being definitive of at least one document property, said data processing system having state assignment functionality for assigning to each document successive ones of said set of states and thus applying said respective property to said instantiation, to thereby successively define for each document within said data processing system a succession of properties during a life cycle of said document.

41. A network based data processing system according to claim 40, wherein each of said document types comprises an assembly of predefined object components.

42. A network based data processing system according to claim 40, comprising document type addition functionality for adding a document type to the document type library during data processing system operation.

43. A network based data processing system according to claim 40, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said set of states includes at least one non-persistent state for attributing to document stages that are not to be saved within the data processing system.

44. A network based data processing system according to claim 43, wherein said at least one non-persistent state is attributable to a document outgoing stage.

45. A network based data processing system according to claim 44, wherein a persistent state is alternately attributable to a document outgoing stage that is to be saved within the data processing system.

46. A network based data processing system according to claim 45, said persistent state being effective to cause generation of said outgoing stage to be delayed.

47. A network based data processing system according to claim 46, wherein said delay is set to depend on system load.

48. A network based data processing system according to claim 40, wherein said set of states includes at least one initiating state for assigning to a document during creation of said document.

49. A network based data processing system according to claim 40, operable to place a document type indicator in a document header.

50. A network based data processing system according to claim 40, wherein at least one of said states has a property of check-in check-out control, to restrict editing level access to a single user at any one time.

51. A network based data processing system according to claim 50, wherein said check-in check-out control comprises a time limitation for any given user.

52. A network based data processing system according to claim 40, operable to set for each data processing system a feature of prompting for saving document history each time any document is assigned another one of said states.

53. A network based data processing system according to claim 40, operable to set for a document type a feature of saving document history each time an instantiation of said document type is assigned another one of said states.

54. A network based data processing system according to claim 40, operable to set for a document a feature of saving document history each time said document is assigned another one of said states.

55. A network based data processing system according to claim 40, comprising owner allocation functionality for allocating an owner to each document.

56. A network based data processing system according to claim 43, said owner allocation functionality being operable to set a feature of placing an indication of said allocated owner in a document header.

57. A network based data processing system according to claim 40, comprising global document identification functionality operable to set a feature of allocating to each document a unique global identifier.

58. A network based data processing system according to claim 57, wherein said global document identification feature is operable to place said unique global identifier for each document in a document header.

59. A network based data processing system according to claim 40, comprising current state indication functionality for setting a feature of placing a current one of said set of states of a document in a document header.

60. A network based data processing system according to claim 40, comprising state ordering functionality for setting for at least one document type a defined order of succession amongst said set of states.

61. A network based data processing system according to claim 40, said state assignment functionality being operable to accept external input to affect assignment of said states.
62. A network based data processing system according to claim 40, said state assignment functionality being operable to affect assignment of said states in accordance with a property of a document.
63. A network based data processing system according to claim 40, said state assignment functionality being operable to alter a property of a document.
64. A network based data processing system according to claim 40, wherein a document security property is definable via said states.
65. A network based data processing system according to claim 64, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said document security property comprises a requirement for a document to acquire at least one digital signature when passing through at least one predetermined stage.
66. A network based data processing system according to claim 65, wherein said digital signature is acquirable from a user.

67. A network based data processing system according to claim 65, wherein said digital signature is acquirable from said data processing system.

68. A network based data processing system according to claim 64, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said document security property.comprises a requirement for a document to have at least one digital signature from a previous stage before being assigned a state corresponding to a new stage.

69. A network based data processing system according to claim 68, wherein said digital signature is acquirable from a user.

70. A network based data processing system according to claim 68, wherein said digital signature is acquirable from said data processing system.

71. A network based data processing system according to claim 64, wherein said data processing system is operable to provide functionality for defining said security property at a document type level.

72. A network based data processing system according to claim 55, wherein a property definable via said states comprises document access permission.

73. A network based data processing system according to claim 55, wherein said owner is exchangeable with a change in state.

74. A network based data processing system according to claim 40, wherein a property definable via said states comprises document access permission.

75. A network based data processing system according to claim 74, wherein document access permission is definable for a given user.

76. A network based data processing system according to claim 75, wherein said access permission comprises permission to delete a document.

77. A network based data processing system according to claim 75, wherein said access permission comprises permission to read a document.

78. A network based data processing system according to claim 75, wherein said access permission comprises permission to change a property of a document.

79. A network based data processing system according to claim 40, wherein a document comprises a header for recording the type and state of said document, and a body for recording additional properties.

80. A network based data processing system according to claim 79, wherein a document further comprises a signature section for recording at least one digital signature.

81. A network based data processing system according to claim 79, wherein a document further comprises a history section for recording document history.

82. A network based data processing system according to claim 79, wherein a document further comprises a relationships section for recording at least one reference to another document.

83. A network based data processing system according to claim 40, wherein a document is presented in one of a group of formats comprising: text, XML, SGML, HTML, dynamic HTML, and VRML.

84. A network based data processing system according to claim 40, wherein said data processing system further comprises a kernel having functionality for providing access to said documents through a standard interface.

85. A network based data processing system according to claim 84, wherein said kernel is operable to provide said access to said documents through said standard interface by receiving an external call to said document through said standard interface, determining a document type of said document,

transforming said external call into a call appropriate to said document type, and forwarding said converted call to said document.

86. A network based data processing system according to claim 84, wherein a protocol of said external call to said kernel includes one of a group comprising: web service protocols, SOAP, DCOM, HTTP, and HTTPS.

87. A network based data processing system according to claim 40, wherein said data processing system further comprises a runtime environment having functionality for providing runtime services for controlling and monitoring said data processing system.

88. A method of providing a development tool for a network based data processing system, the method comprising:
providing a component data store comprising a plurality of object components;
and,
providing a document type developer for assembling document types from said object components and for associating with each a set of states, each state being definitive of at least one document property, said document types being usable to instantiate documents to form a data processing system, said document types being usable with a data processing system having state assignment functionality for assigning to each instantiation successive ones of said set of states and thus applying said respective property to said instantiation, to

thereby successively define for said document type instantiation within said data processing system a succession of properties during a life cycle of said document type instantiation.

89. A method of providing a development tool for a network based data processing system according to claim 88, wherein a protocol of an object component is one of a group of protocols comprising: ActiveX, COM, and COM+.

90. A method of providing a development tool for a network based data processing system according to claim 88, further comprising providing a code inserter for attaching application specific code to said object components.

91. A method of providing a development tool for a network based data processing system according to claim 88, wherein providing a document type developer further comprises providing a code generator for generating application specific code from a set of data provided by a data processing system developer.

92. A method of providing a development tool for a network based data processing system according to claim 88, wherein said code conforms to a coding standard.

93. A method of providing a development tool for a network based data processing system according to claim 88, wherein said code conforms to a naming convention.

94. A method of developing a data processing system, the method comprising:

generating document types from prestored object components; and,
associating with each document type a set of states, each state being definitive of at least one document property, said document types being usable to instantiate documents to form a data processing system having state assignment functionality for assigning to said document type instantiation successive ones of said set of states and thus applying said respective property to said document type instantiation, to thereby successively define for said document type instantiation within said data processing system a succession of properties during a life cycle of said document type instantiation.

95. A method of developing a data processing system according to claim 94, wherein assembling a document type from prestored object components comprises:

inserting application specific code into an object component thereby creating an application component; and,
combining said application component with at least one other component thereby to form an application specific document type.

96. A method of developing a data processing system according to claim 94, wherein assembling a document type further comprises:

defining a set of transitions between pairs of states from said set of states;

defining a set of transition rules comprising at least one transition rule for each of said transitions, said transition rule identifying a condition at which an instantiation of said document type undergoes said transition; and,

associating said set of transitions and said set of transition rules with said document type;

thereby to define a legal succession of properties during a life cycle of a document type instantiation.

97. A method of developing a data processing system according to claim 94, wherein said condition comprises receiving a predetermined external input to a document type instantiation.

98. A method of developing a data processing system according to claim 94, wherein said condition comprises a property of a document type instantiation equaling a predetermined value.

99. A method of developing a data processing system according to claim 96, wherein defining said set of transition rules comprises defining an order of said set of states.

06933756 121626
100. A method of developing a data processing system according to claim 96, wherein each successive state implementation comprises a stage in said document life cycle, and wherein a transition rule comprises a requirement for a document type instantiation to acquire at least one digital signature when passing through a respective stage.

101. A method of developing a data processing system according to claim 100, wherein said digital signature is acquirable from a user.

102. A method of developing a data processing system according to claim 100, wherein said digital signature is acquirable from said data processing system.

103. A method of developing a data processing system according to claim 94, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said set of states includes at least one non-persistent state for attributing to document type instantiation stages that are not to be saved within the data processing system.

104. A method of developing a data processing system according to claim 103, wherein said at least one non-persistent state is attributable to a document type instantiation outgoing stage.

105. A method of developing a data processing system according to claim 104, wherein a persistent state is alternately attributable to a document type instantiation outgoing stage that is to be saved within the data processing system.

106. A method of developing a data processing system according to claim 105, said persistent state being effective to cause generation of said outgoing stage to be delayed.

107. A method of developing a data processing system according to claim 106, wherein said delay is set to depend on system load.

108. A method of defining the processing of a document in a data processing system having state assignment functionality, said document comprising an instantiation of a document type, said document type having a set of states, each state being definitive of at least one document property, for assigning to said document successive ones of said set of states and thus applying said respective property to said document, to thereby successively define for said document a succession of properties during a life cycle of said document, said method comprising:
defining a set of transitions between pairs of states from said set of states;

defining a set of transition rules comprising at least one transition rule for each of said transitions, said transition rule identifying a condition at which an instantiation of said document type undergoes said transition; and, associating said set of transitions and said set of transition rules with said document type.

109. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, wherein said condition comprises an external input to a document type instantiation.

110. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, said wherein said condition comprises a property of a document type instantiation.

111. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, wherein said defining a set of transition rules comprises defining an order of said set of states.

112. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, wherein each successive state implementation comprises a stage in said document life cycle, and wherein a transition rule comprises a requirement for a document

type instantiation to acquire at least one digital signature when passing through a respective stage.

113. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 112, wherein said digital signature is acquirable from a user.

114. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 112, wherein said digital signature is acquirable from said data processing system.

115. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, wherein each successive state implementation comprises a stage in said document life cycle, and wherein said set of states includes at least one non-persistent state for assigning to document type instantiation stages that are not to be saved within the data processing system.

116. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 115, wherein said at least one non-persistent state is assignable to a document type instantiation outgoing stage.

117. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 116, wherein a persistent state is alternately assignable to a document type instantiation outgoing stage that is to be saved within the data processing system.

118. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 117, said persistent state being effective to cause generation of said outgoing stage to be delayed.

119. A method of defining the processing of a document in a data processing system having state assignment functionality according to claim 108, further comprising providing external access to a document through a standard interface by:

receiving a call to a document through said standard interface;
determining a document type of said document;
converting said external call into a call that uses a protocol appropriate to said document type; and,
forwarding said converted call to said document.